PRÁCTICA 3: INTRODUCCIÓN A LA CAPTURA DE ESQUEMAS Y A LA COMPILACIÓN CON QUARTUS II V. 9.0

OBJETIVOS

En esta práctica se empieza a utilizar una herramienta *software* para diseñar *hardware*, comúnmente conocidos como EDA (*Electronic Design Automation*). Concretamente se va a utilizar el programa *Quartus II*, de la empresa *Altera*.

Al finalizar la práctica, el alumno debe ser capaz de:

- Diseñar circuitos digitales usando la herramienta *Quartus II*,
- Entender la importancia del diseño jerárquico.

MATERIAL

- Ordenador personal con *Quartus II*,
- Esquema del circuito (ver Anexo A).

DURACIÓN

1 sesión.

TRABAJO PREVIO

Leer el *Manual de Usuario de la Placa de Lógica Programable DE1*[1] para familiarizarse con la tarjeta de desarrollo de lógica programable del laboratorio.

Leer el tutorial de Quartus II "*Quartus II Introduction Using Schematic Design*",[2] que es un breve manual de usuario de la herramienta para familiarizarse con ella. El alumno puede descargar la herramienta de la página web de Altera (www.altera.com) e instalársela en casa para empezar a trabajar con ella.

INTRODUCCIÓN

Hasta el momento se han estado diseñando pequeños circuitos lógicos que posteriormente se han montado en la tarjeta del laboratorio conectando entre sí varios circuitos integrados. En esta práctica se va a hacer lo mismo pero con una diferencia: en vez de montar el circuito se

---

[1] Disponible en http://www.iit.upcomillas.es/carlosrg/Docencia/LED/DE1_UserManual_v1018.pdf
[2] Disponible en http://www.iit.upcomillas.es/carlosrg/Docencia/LED/tut_quartus_intro_schem.pdf

va a dibujar con un programa de ordenador, y en la práctica siguiente dicho circuito se "descargará" en un dispositivo programable donde se verificará su funcionamiento.

Esto puede parecer un poco raro, pero tiene mucho sentido. La herramienta *Quartus II*, de la casa *Altera*, es una herramienta *software* que permite dibujar (también se suele decir "capturar") esquemas de circuitos lógicos. Esto significa que, a la hora de hacer un circuito, no tenemos más que poner las puertas lógicas que necesitemos en el área de trabajo del programa y conectarlas con líneas que simulan cables.

Estos esquemas o circuitos se pueden hacer en una sola hoja, lo que puede ser un poco incómodo de interpretar si el circuito es muy grande, o puede tener una **estructura jerárquica** (véase la Figura 1), que es una potente cualidad de este tipo de herramientas y que soluciona el problema de grandes esquemas.

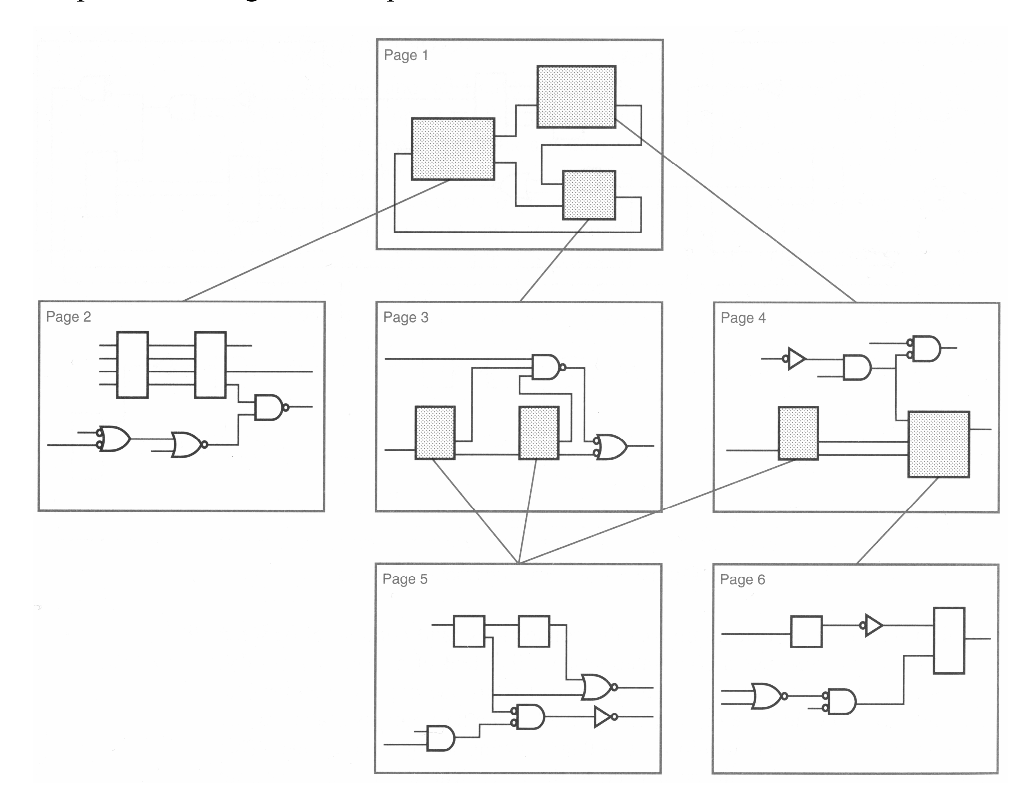


Figura 1. Estructura de un esquema jerárquico.

Para este tipo de esquemas, el nivel superior es una única página que alberga un diagrama de bloques. Normalmente, el nivel superior no contiene ni puertas ni otros elementos lógicos, sino que muestra los subsistemas principales, su interconexión y las entradas y salidas del sistema global. Los subsistemas están definidos en páginas de niveles inferiores, las cuales pueden contener descripciones con puertas lógicas e incluso ellas mismas pueden definir jerarquías de niveles inferiores.

Si una jerarquía particular de bajo nivel se necesita más de una vez, se puede reutilizar varias veces por las páginas de niveles superiores, tal como ocurre con la página 5 del esquema de la Figura 1, en las páginas 3 y 4.

Una vez que el esquema está capturado, se compila, y con el resultado de la compilación se configura un **dispositivo de lógica programable** (PLD – *Programmable Logic Device*), aunque este paso se verá en la próxima práctica.

Estos dispositivos son circuitos integrados, como los que se han manejado en las prácticas anteriores, pero con muchísimas más puertas lógicas, del orden de miles e incluso millones en un solo circuito integrado.

Hay varias familias de dispositivos de lógica programable, como por ejemplo las PLA (*Programmable Logic Array*), los EPLD (*Erasable PLD*), los CPLD (*Complex PLD*), o las FPGA (*Field Programmable Gate Array*). Aunque este texto se centra en las FPGA, que son los que se utilizan en el laboratorio.

Estos dispositivos, las FPGA, se construyen a partir de matrices de lógica programable (*LAB – Logic Array Block*) que se interconectan entre sí mediante una matriz programable de conexiones (véase la Figura 2)

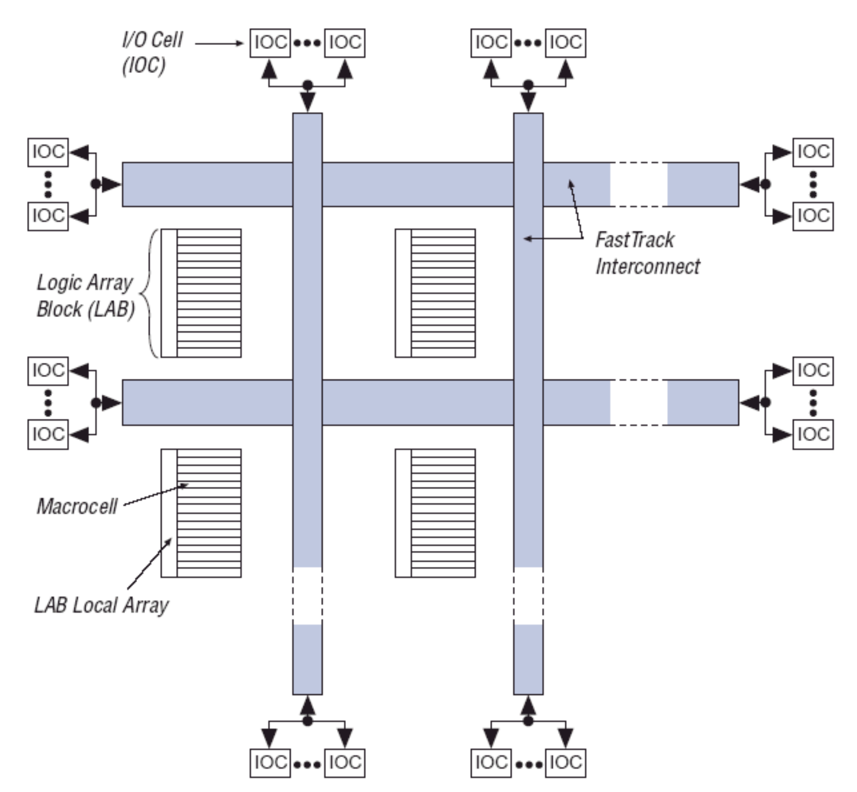


Figura 2. Diagrama de bloques de un CPLD.

Las interconexiones del diseño lógico se almacenan en una memoria no volátil, que puede estar dentro (como por ejemplo en los CPLD) o fuera del dispositivo (como por ejemplo en las FPGA), y que permite mantener la configuración del dispositivo aunque se apague el sistema.

Cada LAB de la FPGA se divide en varias *Macroceldas*, cada una de las cuales está formada por varios elementos de lógica combinacional y uno o dos elementos de memoria (*flip-flop*). La interfaz de señales entre la FPGA y el exterior se realiza mediante las celdas de entrada / salida (*IOC – I/O Cell*) que están conectadas a la matriz de conexiones.

A la hora de trabajar con estos dispositivos hay que capturar el esquema con una herramienta *software* (Quartus II, en nuestro caso), compilarlo para que el programa genere un fichero de configuración del dispositivo y volcarlo al dispositivo (FPGA, en nuestro caso) con la ayuda de un cable. Estos pasos son los que se van a realizar en esta práctica y en la siguiente.

DESARROLLO PRÁCTICO

Para mejorar el orden de los archivos del alumno es necesario crear un directorio de trabajo en donde se almacenarán todos los archivos de esta práctica. Por ejemplo, se puede crear el directorio `C:\Temp\Practica`.

A continuación se detallan los pasos a seguir para capturar y compilar un esquema con *Quartus II*. En la próxima práctica se procederá a la simulación del circuito y a la configuración del dispositivo de lógica programable.

RESUMEN

En esta práctica se va a crear un componente (`Bina7Seg`) que, a partir de un número de 4 bits en binario natural, genere una representación hexadecimal de dicho número en un display de siete segmentos, tal como se muestra en la Figura 3. Este componente se usará en prácticas sucesivas para representar, en un display de 7 segmentos, el valor hexadecimal de un número binario de 4 bits. En la Figura 4 se muestra el esquema general del circuito.

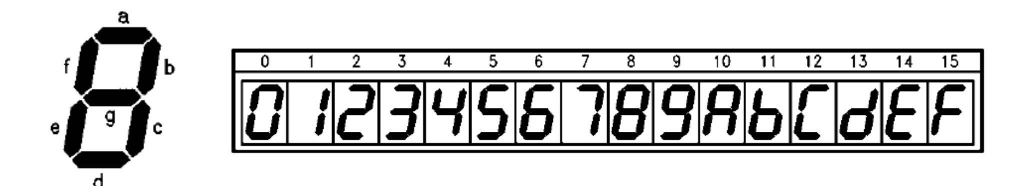


Figura 3. Representación hexadecimal en display de siete segmentos.

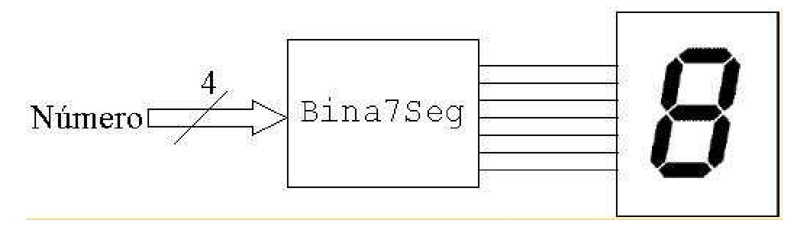


Figura 4. Esquema general.

ARRANCAR QUARTUS II

Lo primero de todo hay que arrancar el programa que se va a utilizar. Para ello hay que elegir la opción `Quartus II 9.0 (32-Bit)` que está en la ruta `Botón de Inicio` -> `Programas` -> `Ingeniería` -> `Altera` -> `Quartus II 9.0`.

DECLARAR UN PROYECTO

Posteriormente se va a declarar **el proyecto** de trabajo en el programa *Quartus II*. Con este proyecto se le va a indicar al programa cuál es el directorio de trabajo y cuál es el archivo principal de la jerarquía (página 1 en el esquema de la Figura 1).

Si al arrancar *Quartus II* aparece la ventana de la Figura 5, hay que pulsar el botón `Sí`, pues lo que se quiere es crear el proyecto. Si no aparece dicha ventana, hay que seleccionar la opción `New Project Wizard`..., en el menú `File`.

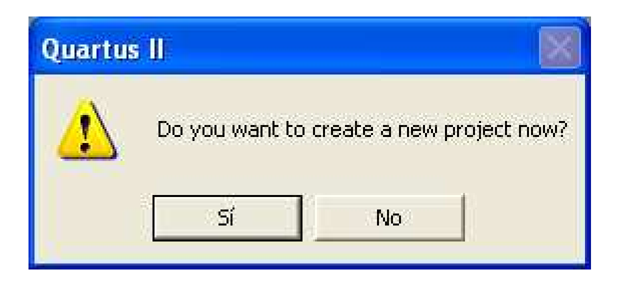


Figura 5. Ventana al arrancar *Quartus II*.

A continuación, después de pulsar `Next`, aparecerá la ventana de la Figura 6, en la que hay que indicar el directorio en el que se va a trabajar (`C:\Temp\Practica`); el nombre del proyecto (`Pract3`); y el nombre del archivo principal de la jerarquía (`Pract3`), tal y como se muestra en la Figura 6. Para evitar confusiones se da el mismo nombre en los dos campos. A continuación pulsar `Next`.

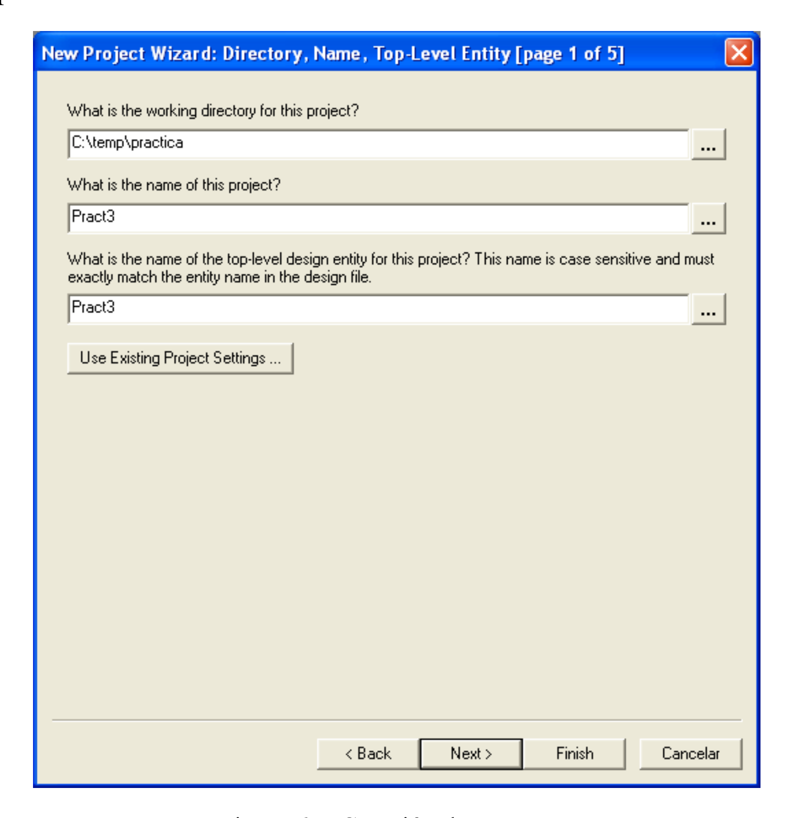


Figura 6. Creación de proyecto.

En la siguiente ventana (véase la Figura 7) no hay que hacer nada. Simplemente pulsar en `Next`.

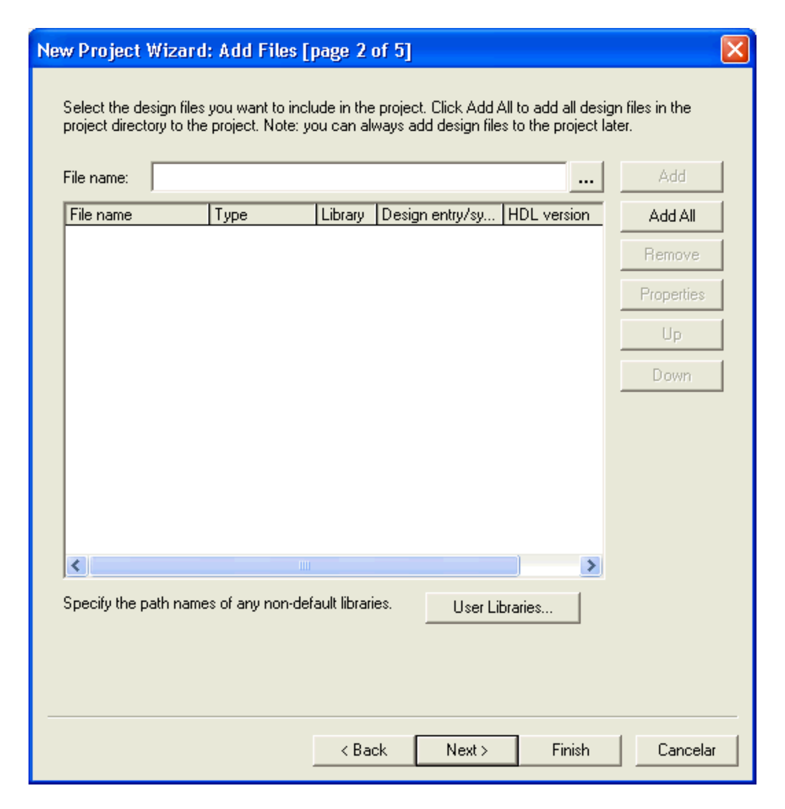


Figura 7. Archivos del proyecto.

En la siguiente ventana (mostrada en la Figura 8) hay que indicarle al programa el dispositivo (familia y modelo) con el que se va a trabajar más adelante. Para ello primero hay que seleccionar la familia `Cyclone II`, y después, en la lista de dispositivos disponibles de la parte de abajo, seleccionar el dispositivo `EP2C20F484C7`, que es el que hay montado en la tarjeta de desarrollo de lógica programable del laboratorio. Con estos pasos ya se tiene todo lo necesario para declarar el proyecto, por lo que, para terminar, hay que pulsar el botón `Finish`.

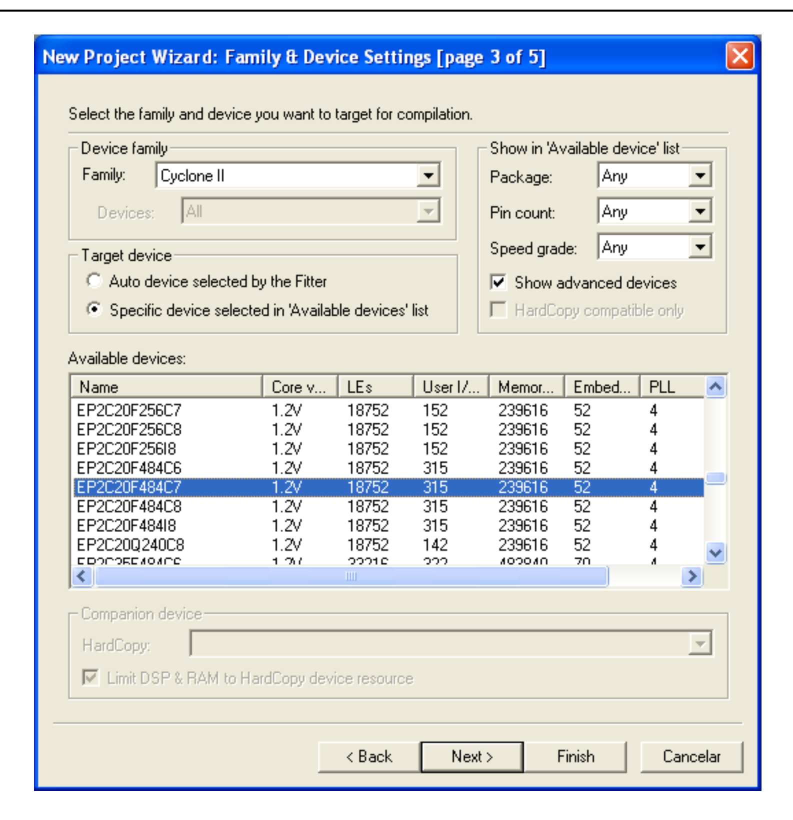


Figura 8. Asignación del dispositivo.

CAPTURAR EL CIRCUITO CONVERSOR DE BINARIO A SIETE SEGMENTOS

CREAR UN ARCHIVO DE CAPTURA

A continuación se va a crear un esquema nuevo que contendrá el circuito del componente. Seleccionando la opción `New...` en el menú `File` aparece la ventana de la Figura 9, que permite elegir el tipo de archivo que se va a crear.

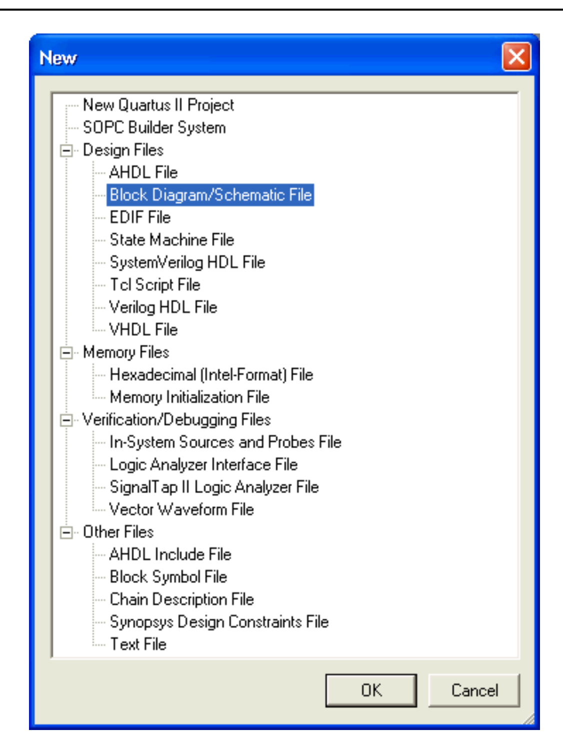


Figura 9. Selección del tipo de archivo a crear.

Existen cuatro grupos diferentes de archivos. Archivos de Diseño, archivos de Memoria, archivos de Verificación y otros tipos de archivos complementarios. En este curso se van a utilizar principalmente los siguientes: archivo gráfico (*Block Diagram/Schematic File*) para capturar un esquema; archivo de símbolo (*Block Symbol File*) para un componente; y archivo de forma de onda (*Vector Waveform File*), para describir la evolución temporal de las entradas con vistas a la simulación de un circuito previamente creado.

Como lo que se desea es capturar un esquema, hay que seleccionar la opción `Block Diagram/Schematic File` en la ventana anterior. El archivo creado tendrá extensión `.bdf`.

Después de pulsar `OK` aparece la ventana de captura de esquemas, que es una hoja en blanco donde se va a dibujar el esquema. Antes de nada conviene guardar el archivo en el directorio de trabajo con el nombre `Bina7Seg.bdf`, mediante la opción `Save As`... del menú `File`.

Aparecerá entonces la ventana de la Figura 10, en la que hay que indicar el nombre del archivo (`Bina7Seg.bdf`). Tenga cuidado con el nombre, pues, por defecto, aparece el nombre del proyecto (`Pract3`). Es muy importante que la opción `Add file to current project` (en la parte inferior de la ventana) esté activada. De esta forma se le está diciendo al programa que el archivo `Bina7Seg.bdf` pertenece al proyecto actual (`Pract3`), para que lo tenga en cuenta en el momento de compilar (tarea que se realizará más adelante).

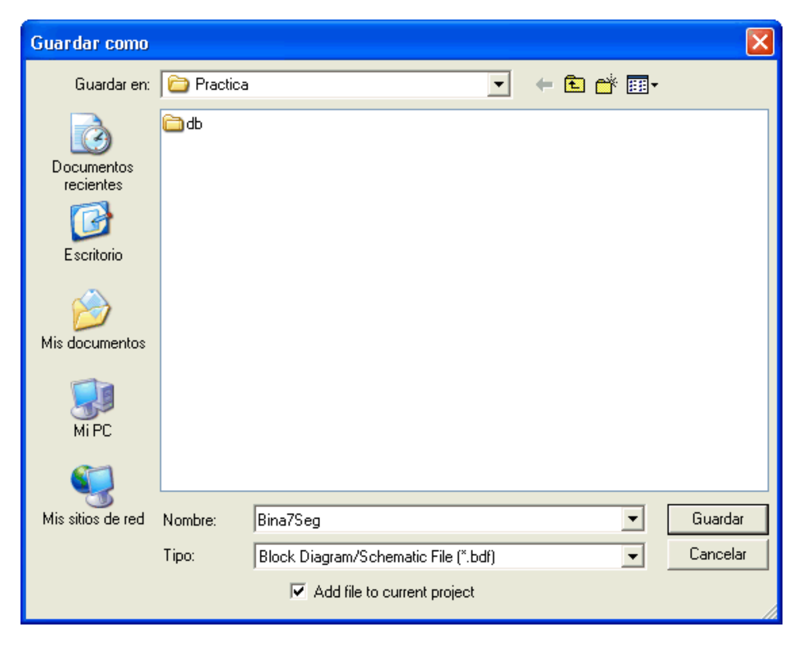


Figura 10. Guardar un archivo.

INTRODUCIR COMPONENTES EN LA VENTANA DE CAPTURA DE ESQUEMAS

A continuación hay que dibujar el esquema mostrado en el Anexo A de esta práctica.

Para introducir un componente en la ventana de captura hay que pulsar el botón , que está en la barra de botones de la izquierda, o hacer doble clic con el ratón en la hoja donde se desea insertar, tras lo cual se presentará una ventana como la mostrada en la Figura 11. De esta forma se le puede decir al programa qué componente, de los que dispone en su base de datos, se quiere introducir.

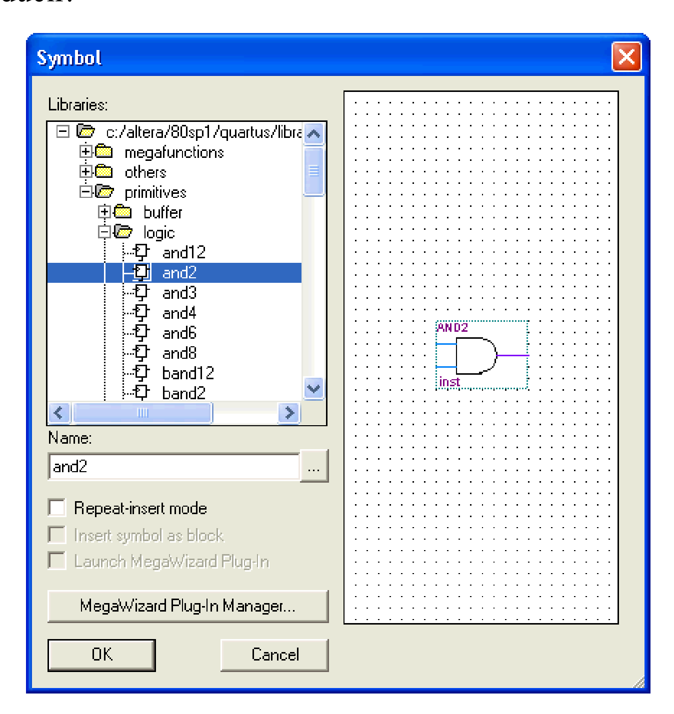


Figura 11. Selección de componentes.

En esta ventana se muestran las librerías disponibles. Estas librerías no son más que directorios dentro del ordenador donde se encuentran una serie de símbolos que se pueden usar en los circuitos diseñados. Las librerías que aparecen son `primitives` (primitivas), que contiene los componentes básicos de diseño: puertas lógicas, entradas, salidas, flip-flops, etc.; `megafunctions`, que son componentes complejos diseñados por Altera a partir de los componentes básicos; y `others`, que son otras librerías con funciones configurables y circuitos comerciales.

Para seleccionar el componente que se desea introducir, existen dos alternativas: teclearlo directamente en la casilla `Name` (siempre que se conozca el nombre), o seleccionarlo en la ventana `Libraries`, buscando en la librería adecuada.

En primer lugar se va introducir el componente `input`, que representa una señal de entrada al circuito de la hoja activa. Teclear su nombre en el campo `Name` y pulsar en `OK` para introducir el componente. Como puede comprobarse, el componente ha aparecido en la ventana de dibujo y hay que hacer clic en el lugar donde se quiera insertarlo.

Si no gusta dónde ha quedado, se puede arrastrar con el ratón o usar la técnica de cortar y pegar, la cual es muy útil en su versión copiar y pegar para introducir componentes iguales que ya estén insertados en la hoja, sin necesidad de pasar por la ventana de selección de componentes.

Una vez situado el puerto de entrada en su sitio, hay que introducir el componente `not`, situándolo por debajo y a la derecha del anterior, como se muestra en la Figura 12.

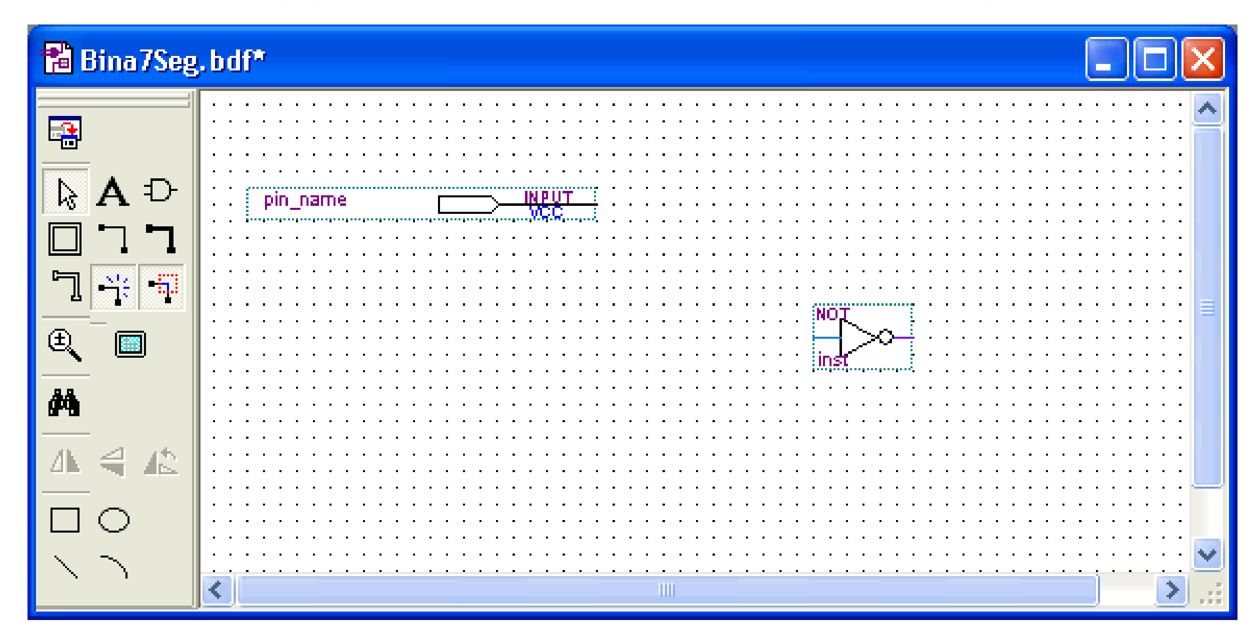


Figura 12. Inserción de componentes.

CABLEAR

Para realizar el cableado hay que acercar el cursor del ratón a una patilla del componente y, cuando la flecha cambie a una cruz ( ), arrastrar el cable a la patilla del otro componente, sin soltar el botón del ratón, salvo que se desee realizar más de un codo, en cuyo caso basta con soltar y volver a pulsar el botón del ratón, lo que permitirá hacer un nuevo codo.

Obsérvese, en la Figura 13, que la unión de dos cables se representa por un punto grueso.

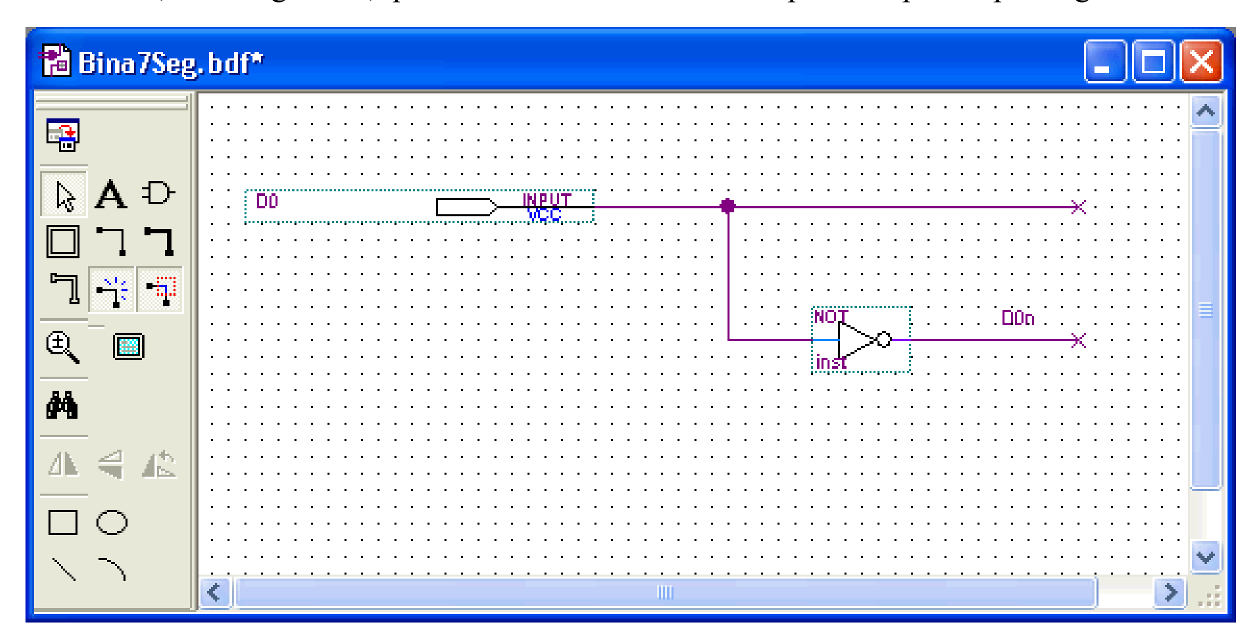


Figura 13. Cableado y etiquetado.

ASIGNAR NOMBRES

Una vez finalizado el cableado de estos dos componentes como se muestra en la Figura 13, se va a dar un nombre al puerto de entrada (componente `input`) y a la salida del inversor (puerta **NOT**).

El primer nombre servirá para identificar el puerto de entrada cuando se use el símbolo del circuito en otro esquema.

La razón de dar un nombre a la salida del inversor es la de ahorrar tiempo en cablear componentes y mejorar la legibilidad del esquema, pues en los programas de captura de esquemas, para conectar dos elementos, se puede o bien llevar un cable de uno a otro, tal como se acaba de hacer entre el puerto de entrada y el inversor, o bien darle el mismo nombre a dos cables dentro del circuito, lo que emulará una conexión entre componentes.

Para cambiar el nombre del puerto de entrada de `pin_name` a `D0` hay que hacer doble clic encima de `pin_name` y cambiar el nombre a continuación. Si se hace doble clic en el símbolo `input`, aparece la ventana de la Figura 14, donde también se puede introducir el nombre.

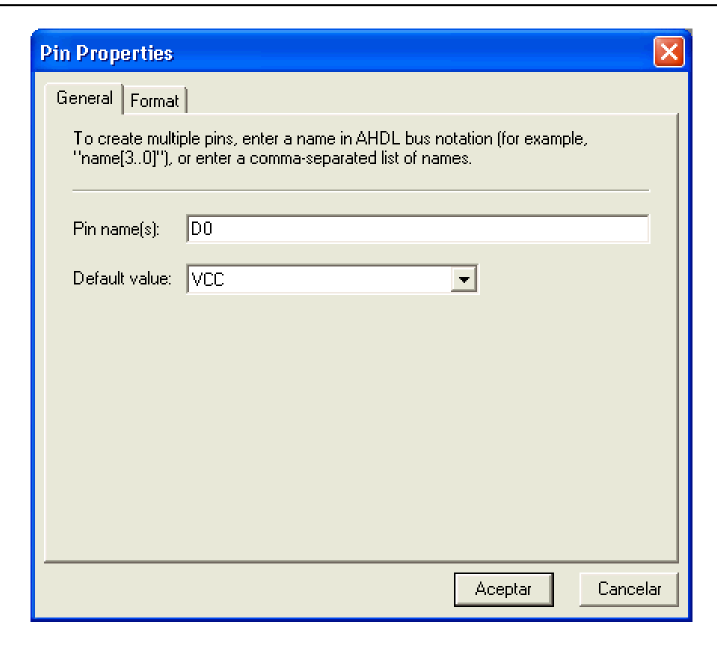


Figura 14. Asignación de nombre.

Para nombrar la salida de la puerta inversora, en primer lugar es necesario conectar un trozo de cable, tal como se muestra en la Figura 13, y a continuación, haciendo un clic encima de dicho cable, **éste se pondrá azul** y el cursor cambiará a "*modo texto*", con lo que se podrá introducir el nombre de este cable: `D0n`. Al terminar, si está bien hecho, el cable y el texto deben ser de color granate, y al pinchar en el cable se deben seleccionar el cable (en azul) y el nombre (dentro de una caja azul). Si el texto aparece en verde, el programa lo entenderá como un comentario y no como el nombre del cable.

Ahora hay que repetir los pasos anteriores para dibujar el resto del esquema del Anexo A, sabiendo que todas las puertas pertenecen a la librería `primitive` -> `logic`.

CREAR EL SÍMBOLO DEL ESQUEMA

Una vez capturado todo el esquema del Anexo A, es necesario crear el símbolo asociado al esquema que se acaba de capturar, de forma que se pueda utilizar este componente en otros esquemas.

Esta tarea se puede realizar automáticamente mediante el comando `Create Symbol Files for Current File` que se encuentra en la opción `Create/Update` del menú `File`. Esta acción abrirá la ventana mostrada en la Figura 15, en la que se muestra el nombre del símbolo (`Bina7Seg.bsf`), igual al nombre del esquema, en la que hay que pulsar el botón `Guardar`. Esto creará el símbolo y lo asociará al proyecto en curso avisando con el mensaje de la Figura 16. Pulsar en `Aceptar` y continuar.

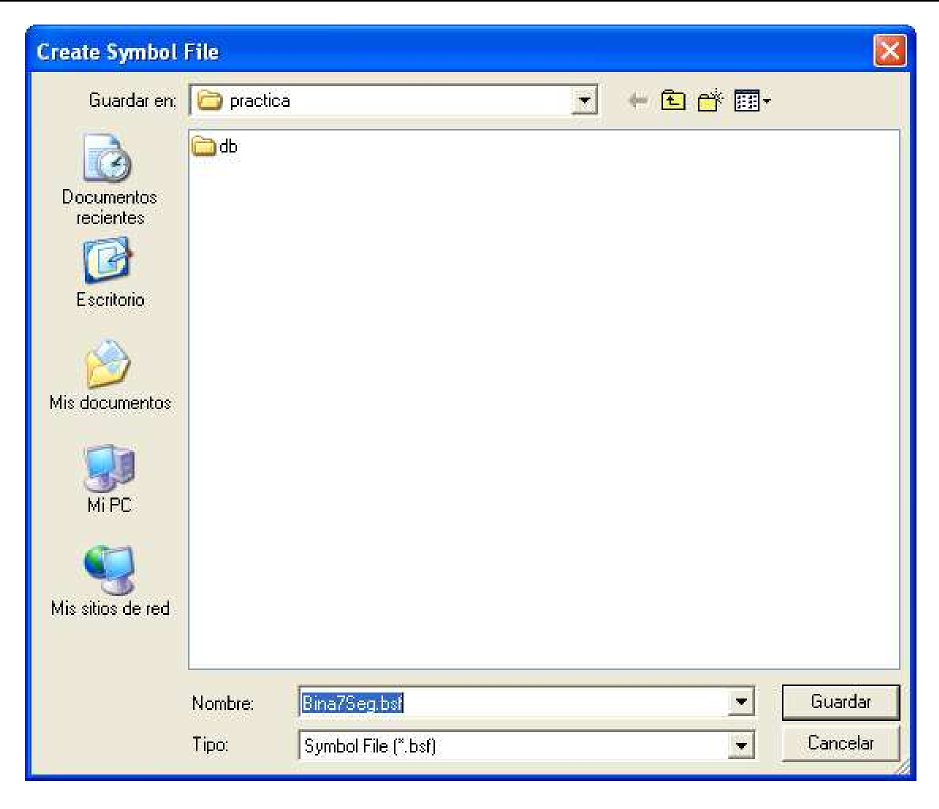


Figura 15. Ventana para guardar un símbolo.

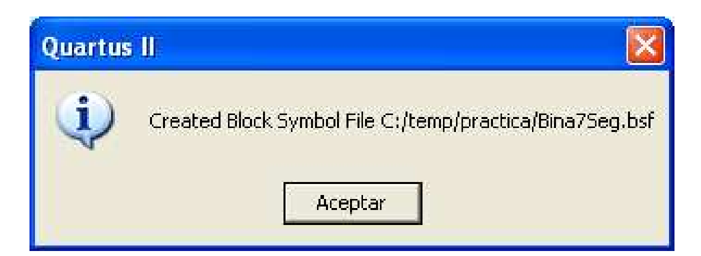


Figura 16. Aviso de que el símbolo se ha creado.

EDITAR UN SÍMBOLO

Para editar el símbolo que se acaba de crear, hay que abrir el archivo `Bina7Seg.bsf` con la opción `Open` del menú `File`. En la ventana que se abre, hay que indicar que el tipo de archivo que se busca es `Graphic Files (*.gdf, *.bdf, *.bsf, *.sym)` y luego indicarle el nombre del archivo con la extensión **`.bsf`** (véase la Figura 17).

Al editar el símbolo, se permite cambiar el orden de las entradas y salidas de la caja, para lo cual basta con **hacer doble clic sobre el nombre de la entrada o salida** y teclear el nuevo nombre. No es correcto arrastrar los nombres de las señales.

El símbolo final ha de quedar como el mostrado en la Figura 18.

Una vez que se tengan ordenadas las entradas y las salidas, se pueden guardar los cambios y cerrar la ventana del editor de símbolos.

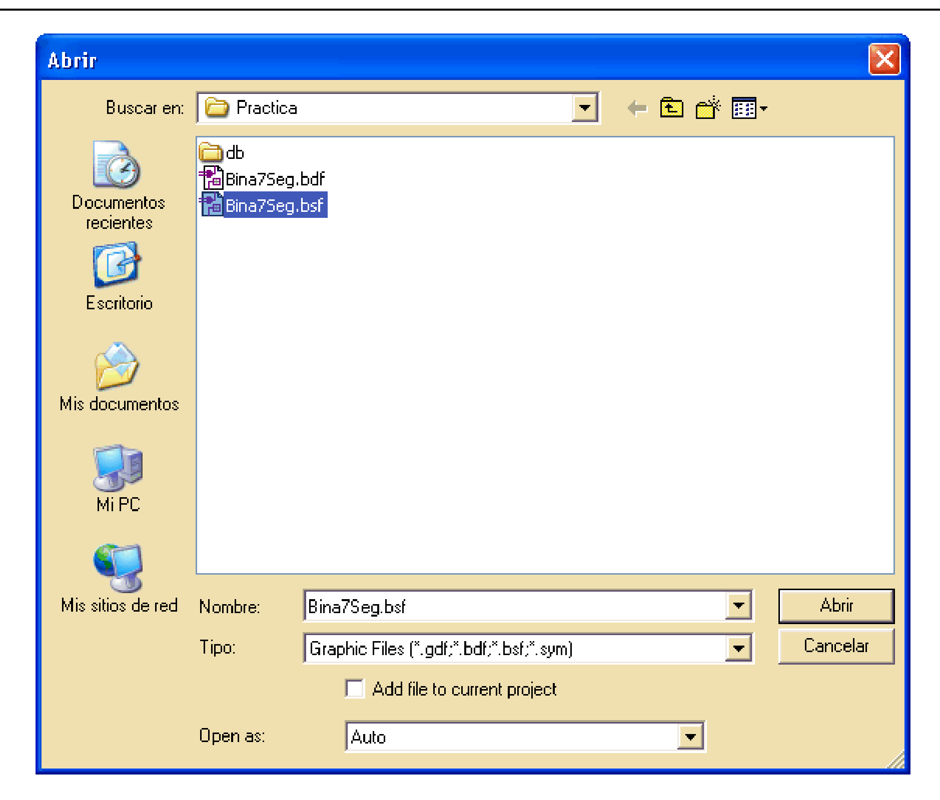


Figura 17. Abrir un archivo de símbolo.

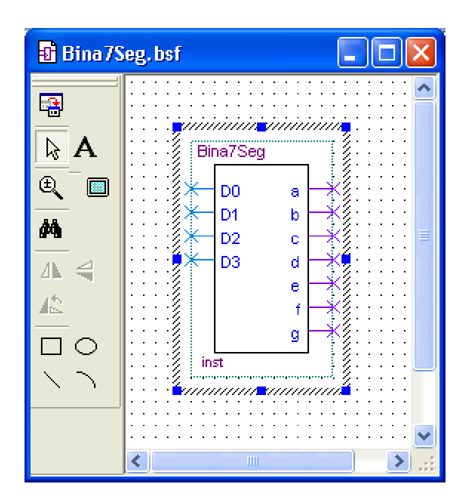


Figura 18. Símbolo del esquema.

ARCHIVO SUPERIOR DE LA JERARQUÍA

Una vez finalizado el componente del Anexo A, se va a realizar un circuito que muestre, en el display `HEX0` de la tarjeta de desarrollo de lógica programable, el número codificado en binario en los interruptores `SW3` al `SW0` (véase el *Manual de Usuario de la Placa de Lógica Programable DE1*[3] para familiarizarse con los componentes de la tarjeta). El esquema final, tras realizar los pasos que se describen más adelante, debe quedar como el que se muestra en la Figura 19.

[3] Disponible en http://www.iit.upcomillas.es/carlosrg/Docencia/LED/DE1_UserManual_v1018.pdf

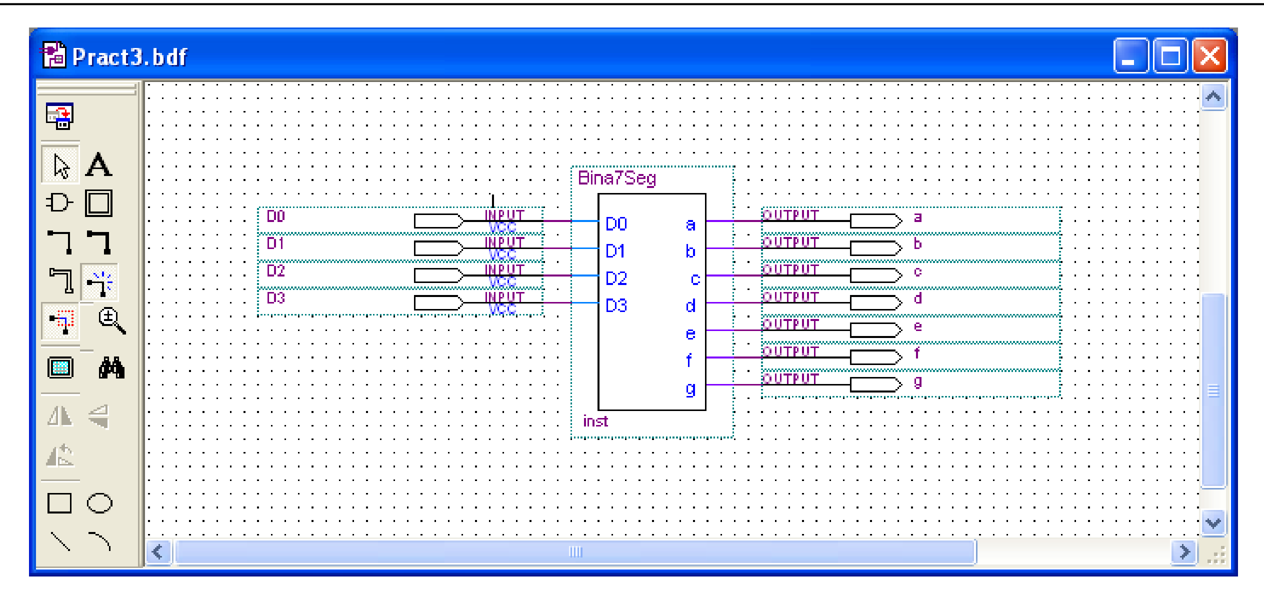


Figura 19. Nivel superior de la jerarquía.

Tal como se puede apreciar, dicho esquema consta tan sólo de cuatro puertos de entrada y ocho puertos de salida que, en este caso (el archivo superior de la jerarquía), simbolizan las entradas y salidas físicas de la FPGA. Estas entradas y salidas se han conectado, en este esquema sencillo, al componente creado anteriormente (`Bina7Seg.bdf`).

La razón por la que no se han puesto las entradas y salidas físicas directamente en el componente `Bina7Seg.bdf` es porque, de esta forma, dicho componente está listo para usarse en cualquier diseño que se desee sin más que insertarlo en la hoja de dibujo y conectar sus terminales. Si no se hubiese creado el componente, cada vez que se necesitara usar un decodificador de binario a siete segmentos se tendría que dibujar de nuevo el esquema del decodificador (Anexo A), que como se ha comprobado, no es nada simple. Viéndolo así, se podría pensar en la técnica de copiar y pegar para cuando hiciese falta, pero ¿qué pasa si en un esquema se necesitan cuatro decodificadores? Se tendrían entonces problemas de espacio y de legibilidad.

En resumen, siempre que se realice un diseño complejo es necesario dividir el circuito en bloques, e implantar cada bloque por separado, en forma de componentes. Esto permite simplificar el diseño en todas sus vertientes. En primer lugar, el proceso de dibujo es más simple al centrarse cada vez en un bloque pequeño, en lugar de en un circuito enorme. En segundo lugar, se puede comprobar el funcionamiento de cada bloque por separado, tanto en simulación como físicamente en la FPGA, lo que permite depurar el circuito mucho más rápida y cómodamente. En tercer lugar, la documentación del circuito es más clara al tener varios esquemas de tamaños manejables, en lugar de una “sábana” con un montón de puertas interconectadas. Por último, si los bloques que se diseñan son lo suficientemente genéricos, como por ejemplo el decodificador de binario a siete segmentos diseñado en esta práctica, se pueden usar en otros diseños sin más que insertarlos.

A este tipo de diseño organizado en bloques se le denomina **<u>diseño jerárquico</u>**, puesto que el circuito está formado por una jerarquía de bloques que parten de un nivel superior, en el que se representa el circuito con un alto nivel de abstracción, para ir aumentando el nivel de detalle conforme se desciende en la jerarquía.

Antes de comenzar el diseño del circuito global, es decir, el nivel superior de la jerarquía, es necesario crear un nuevo esquema con el nombre `Pract3.bdf`. Para ello hay que crear un nuevo archivo de captura de esquemas (`Block Diagram/Schematic File`) y luego guardarlo como `Pract3.bdf`, asegurándose de que la opción `Add file to current project`, en la ventana `Guardar como`, está seleccionada.

Después hay que insertar el componente `Bina7Seg`, de la misma forma que se han insertado las puertas lógicas del Anexo A. En esta ocasión el componente está en la carpeta `Project` de la ventana de símbolos (véase la Figura 20).

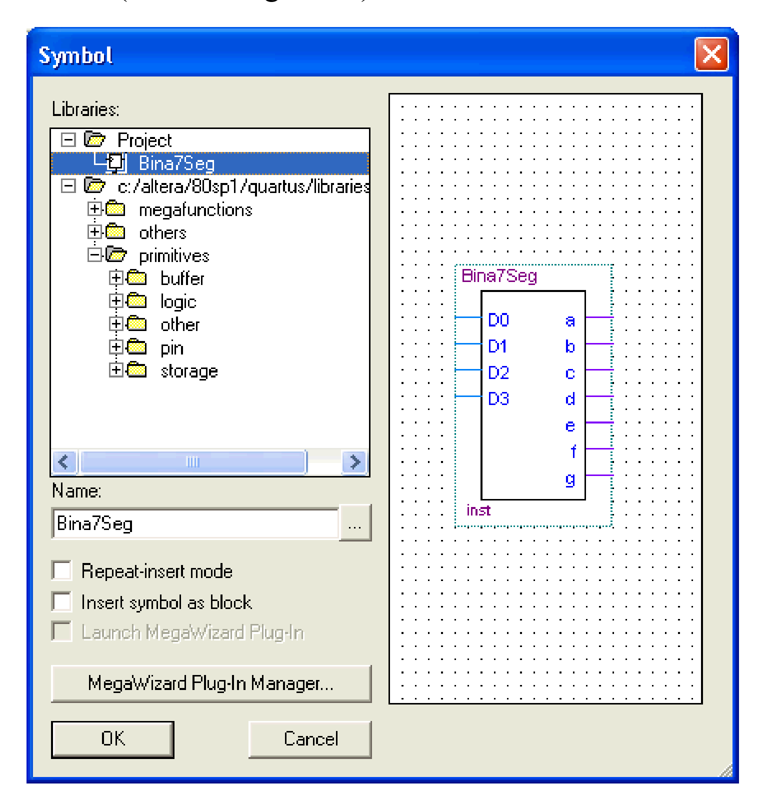


Figura 20. Ventana de componentes.

A continuación hay que insertar los terminales de entrada y salida, y nombrarlos como se muestra en el esquema la Figura 19.

Una vez que se tiene el esquema como el mostrado en la Figura 19, se ha terminado la primera fase del diseño estructural: **la fase de captura**. A continuación se pasará a la segunda fase, compilación del circuito, para depurar posibles errores, y posteriormente se realizarán, en la práctica 4, las fases de simulación del circuito y programación de la FPGA.

Compilar el circuito conversor de binario a siete segmentos

Una vez que se ha terminado con la tarea de capturar todos los esquemas de la jerarquía, es necesario compilar el proyecto, y por tanto todos los archivos de los que consta el mismo, para verificar que el circuito está libre de errores.

COMPILAR EL PROYECTO

Para compilar, primero hay que guardar todos los esquemas, por si se ha hecho algún cambio de última hora, y luego arrancar el Compilador (véase la Figura 21), que se encuentra en la opción `Compiler Tool` del menú `Processing`.

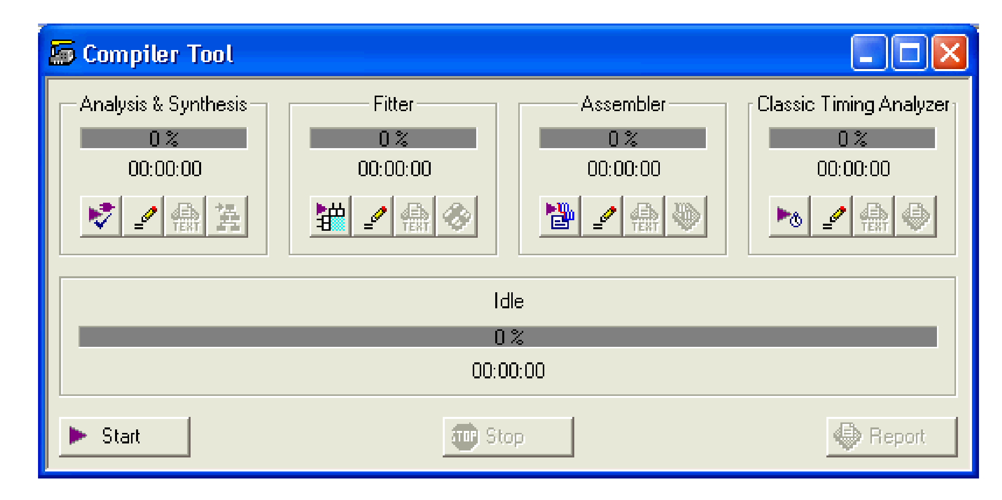


Figura 21. Ventana del compilador.

Para arrancar la compilación hay que pulsar el botón `Start`. Pasados unos instantes terminará el proceso con una ventana en la que se informa del número de avisos (*warnings*) que se han producido en el proceso (véase la Figura 22). Si todo está correctamente sólo deberían aparecer los avisos que se muestran en la Figura 23, además de los mensajes de información del propio programa.

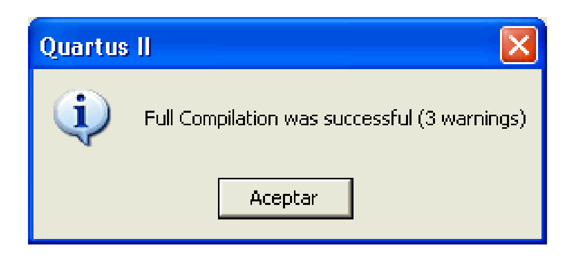


Figura 22. Aviso de fin de compilación.

```
Type    Message
Info: ********************************************************************
Info: ********************************************************************
Info: ********************************************************************
Info: ********************************************************************
Info: Assembler is generating device programming files
Info: Average interconnect usage is 0% of the available device resources
Info: Command: quartus_asm --read_settings_files=off --write_settings_files=off Pract3 -c Pract3
Info: Command: quartus_fit --read_settings_files=off --write_settings_files=off Pract3 -c Pract3
Info: Command: quartus_map --read_settings_files=on --write_settings_files=off Pract3 -c Pract3
Info: Command: quartus_tan --read_settings_files=off --write_settings_files=off Pract3 -c Pract3 --timing_analysis_only
Info: Delay annotation completed successfully
Info: Device migration not selected. If you intend to use device migration later, you may need to change the pin assign
Info: Elaborating entity "Bina7Seg" for hierarchy "Bina7Seg:inst"
Info: Elaborating entity "Pract3" for the top level hierarchy
Info: Finished register packing
Info: Fitter converted 3 user pins into dedicated programming pins
Info: Fitter is performing an Auto Fit compilation, which may decrease Fitter effort to reduce compilation time
Info: Fitter is using the Classic Timing Analyzer
Info: Fitter placement operations beginning
Info: Fitter placement operations ending: elapsed time is 00:00:01
Info: Fitter placement preparation operations beginning
Info: Fitter placement preparation operations ending: elapsed time is 00:00:00
Info: Fitter placement was successful
Info: Fitter preparation operations ending: elapsed time is 00:00:01
Info: Fitter routing operations beginning
Info: Fitter routing operations ending: elapsed time is 00:00:00
Info: Found 1 design units, including 1 entities, in source file Bina7Seg.bdf
Info: Found 1 design units, including 1 entities, in source file Pract3.bdf
Info: Generated suppressed messages file C:/temp/practica/Pract3.fit.smsg
Info: High junction temperature is 85 degrees C
Info: I/O bank details before I/O pin placement
Info: Implemented 18 device resources after synthesis - the final resource count might be different
Info: Longest tpd from source pin "D3" to destination pin "f" is 12.986 ns
Info: Low junction temperature is 0 degrees C
Info: Parallel compilation is enabled and will use 4 of the 4 processors detected
Info: Parallel compilation is enabled and will use 4 of the 4 processors detected
Info: Quartus II Analysis & Synthesis was successful. 0 errors, 0 warnings
Info: Quartus II Assembler was successful. 0 errors, 0 warnings
Info: Quartus II Classic Timing Analyzer was successful. 0 errors, 0 warnings
Info: Quartus II Fitter was successful. 0 errors, 3 warnings
Info: Quartus II Full Compilation was successful. 0 errors, 3 warnings
Info: Running Quartus II Analysis & Synthesis
Info: Running Quartus II Assembler
Info: Running Quartus II Classic Timing Analyzer
Info: Running Quartus II Fitter
Info: Selected device EP2C20F484C7 for design "Pract3"
Info: Started post-fitting delay annotation
Info: Starting register packing
Info: Statistics of I/O pins that need to be placed that use the same VCCIO and VREF, before I/O pin placement
Info: The Fitter performed an Auto Fit compilation.  Optimizations were skipped to reduce compilation time.
Info: Timing requirements not specified -- quality metrics such as performance and power consumption may be sacrificed
Info: Writing out detailed assembly data for power analysis
Warning: Found 7 output pins without output pin load capacitance assignment
Warning: No exact pin location assignment(s) for 11 pins of 11 total pins
Warning: The Reserve All Unused Pins setting has not been specified, and will default to 'As output driving ground'.

System (2) | Processing (55) | Extra Info | Info (52) | Warning (3) | Critical Warning | Error | Suppressed (6) | Flag
Message: 0 of 128    Location:    Locate
```

Figura 23. Ventana de mensajes con los *warning* de la práctica.

Si se produce algún error, aparecerá la ventana de la Figura 24, donde se indica el número de errores detectados, y en la ventana de mensajes se mostrarán los errores producidos (véase un ejemplo en la Figura 25). Es MUY IMPORTANTE **LEER DETENIDAMENTE** el texto del error en la ventana de mensajes, pues indica cuál es el problema. Si se hace doble clic sobre el mensaje de error, el programa abre la ventana donde lo ha localizado y lo resalta. Si no se sabe interpretar el error, pida ayuda al profesor.



Figura 24. Aviso de compilación con errores.

Type	Message
Info	Info: Found 1 design units, including 1 entities, in source file Pract3.bdf
Info	Info: Elaborating entity "Pract3" for the top level hierarchy
Info	Info: Elaborating entity "Bina7Seg" for hierarchy "Bina7Seg:inst"
Error	Error: Illegal name "D0" -- pin name already exists
Error	Error: Can't elaborate user hierarchy "Bina7Seg:inst"
Error	Error: Quartus II Analysis & Synthesis was unsuccessful. 2 errors, 0 warnings
Error	Error: Quartus II Full Compilation was unsuccessful. 4 errors, 0 warnings

System (6) | Processing (11) | Extra Info | Info (7) | Warning | Critical Warning | Error (4) | Suppressed | Flag

Figura 25. Ventana de mensajes con errores.

Una vez que se hayan corregido todos los errores mostrados por *Quartus II*, hay que guardar los cambios realizados y volver a compilar (pulsar el botón `Start` de la ventana `Compiler Tool`). Si aparecen nuevos errores, se han de subsanar y repetir el proceso hasta que compilación sea satisfactoria.

También es muy importante leer los mensajes asociados a los avisos (*warning*) para verificar que no se ha cometido ningún error. Por ejemplo, en esta práctica, el compilador avisa de que existen 11 señales sin asignación exacta de patillas de la FPGA, como se puede observar en la Figura 23. En esta ocasión no hay que preocuparse por este aviso, pues todavía no se han asignado las señales del circuito a las patillas de la FPGA, pero en otras ocasiones puede ser indicio de un error.

VISUALIZAR LA JERARQUÍA

Para Visualizar la Jerarquía del proyecto ha de abrirse la ventana `Project Navigator` que se encuentra en la opción `Utility Windows` del menú `View` (véase la Figura 26).

En este esquema se puede observar que la jerarquía del proyecto está formada por el esquema `Pract3.bdf`, del que cuelga el esquema `Bina7Seg.bdf`.

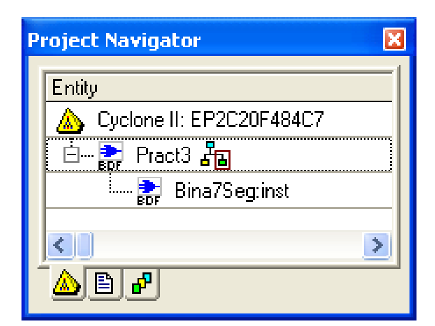


Figura 26. Jerarquía del proyecto.

ASIGNAR PATILLAS DE LA FPGA A TERMINALES DE ENTRADA Y SALIDA DEL ESQUEMA

Una vez que se ha verificado que los esquemas del proyecto están libres de errores, el último paso de esta práctica es la asignación de las patillas físicas del dispositivo de lógica programable (FPGA) a los terminales de entrada y salida del esquema del nivel superior de la jerarquía.

Para ello hay que elegir la opción `Pin Planner` del menú `Assignments`. Aparecerá entonces la ventana de la Figura 27.

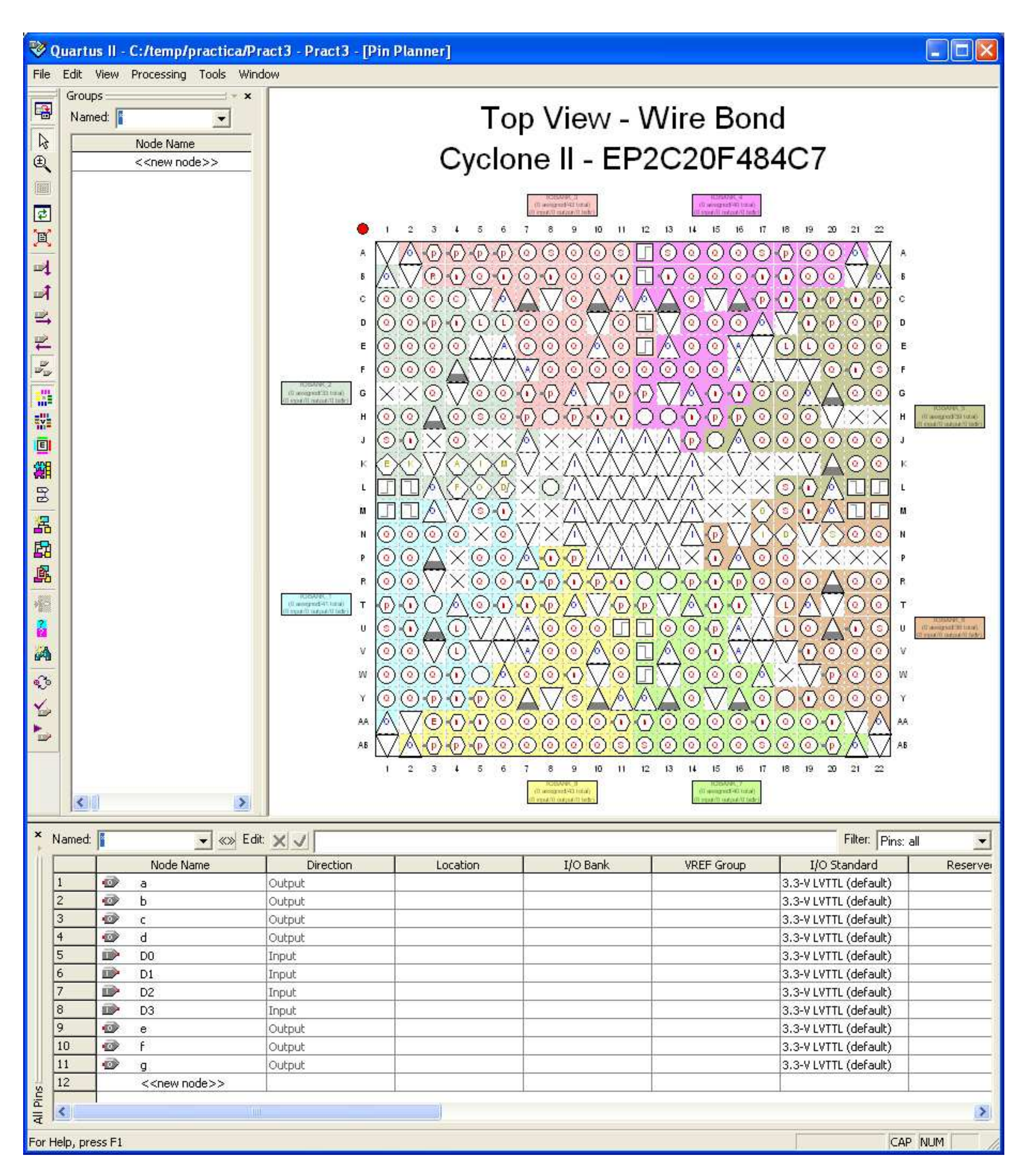


Figura 27. Ventana de asignación de patillas.

Para asignar las patillas de la FPGA a los terminales del esquema se puede hacer de dos formas. La primera consiste en hacer doble clic en la columna `Location` y en la fila del terminal que se quiere asignar, y a continuación elegir el pin correspondiente al terminal de la lista desplegada. Por ejemplo, al terminal ‘`D0`’ le corresponde el pin L22 (véase la Figura 28).

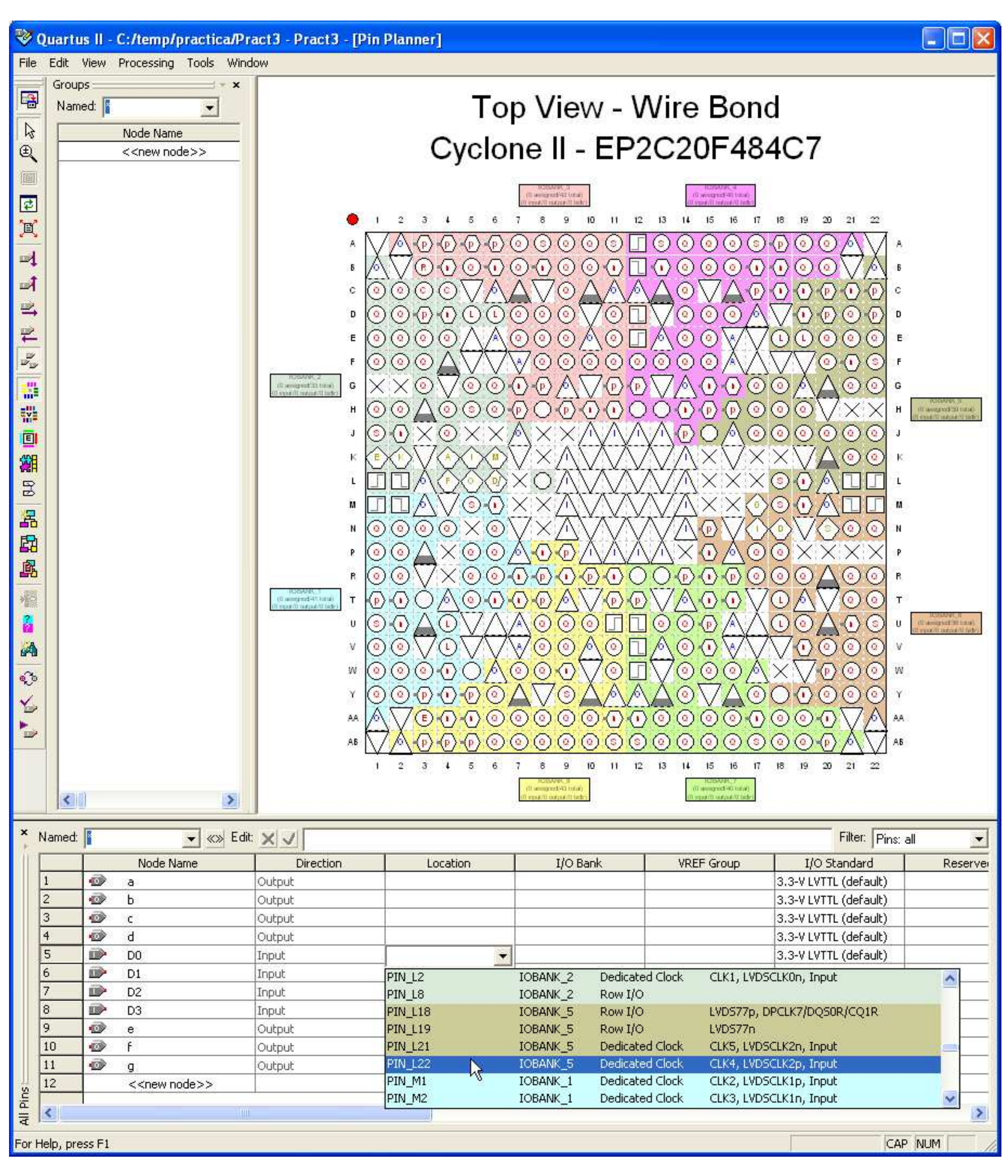


Figura 28. Ventana de asignación de patillas.

La segunda forma consiste, primero, en seleccionar uno de los terminales haciendo clic en la fila deseada dentro de la columna `Node Name`, y, segundo, arrastrar dicho terminal sobre uno de los pines del mapa de la FPGA que hay en la parte superior de la ventana. Por ejemplo, al terminal '`a`' le corresponde el pin J2 (véase la Figura 29).

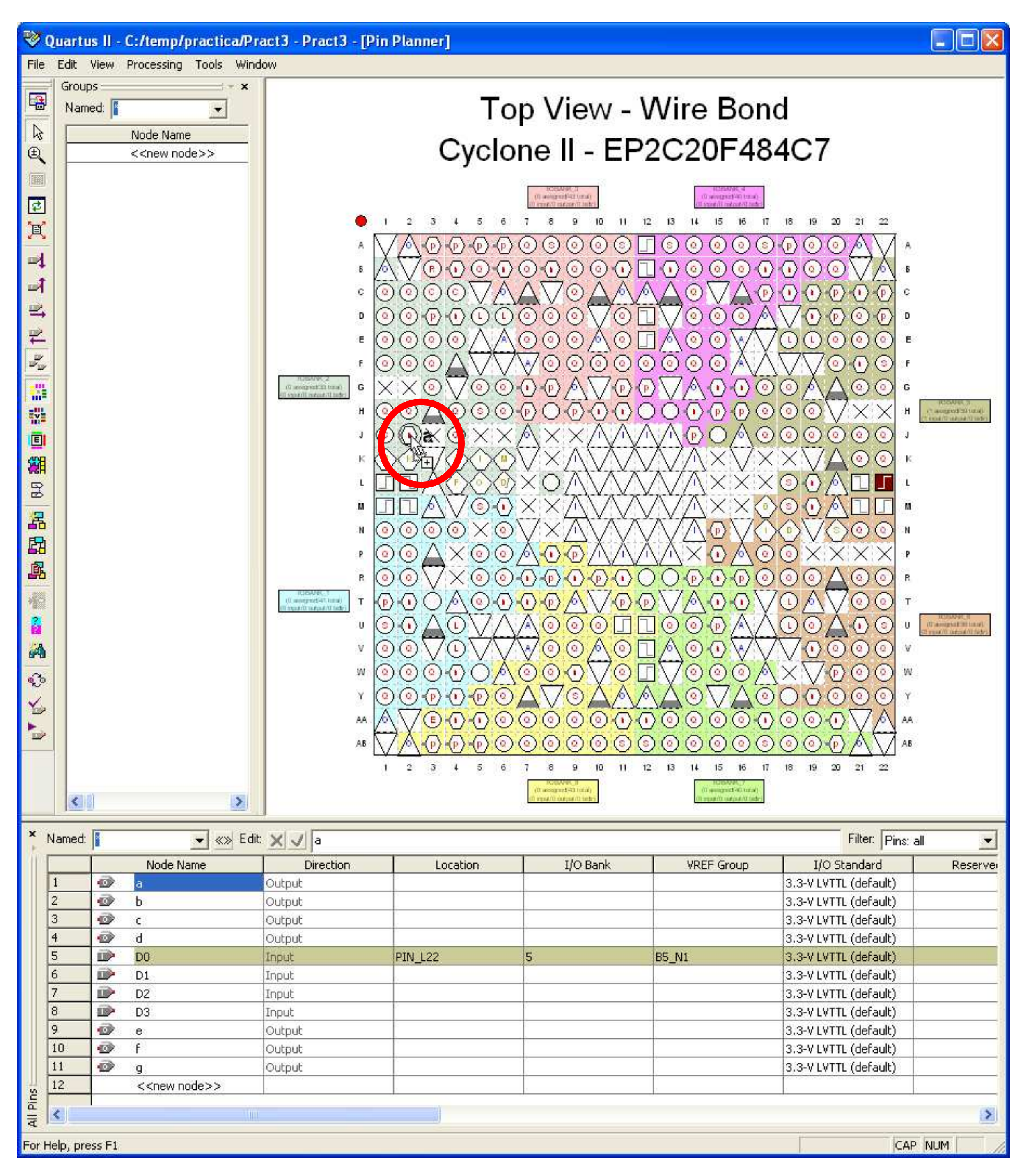


Figura 29. Ventana de asignación de patillas.

Hay que repetir este proceso para el resto de los terminales de entrada y de salida con las asignaciones que se muestran en la Tabla 1. Al finalizar hay que cerrar la ventana del `Pin Planner`, para asegurarse de que guarda las asignaciones, y volver a compilar el proyecto (botón `Start` del `Compiler Tool`) para que tenga en cuenta las asignaciones. El esquema debe quedar como en la Figura 30.

Señal	Tipo	Patilla FPGA	Componente
D0	Entrada	PIN_L22	**SW0**
D1	Entrada	PIN_L21	**SW1**
D2	Entrada	PIN_M22	**SW2**
D3	Entrada	PIN_V12	**SW3**
a	Salida	PIN_J2	**HEX0[0]**
b	Salida	PIN_J1	**HEX0[1]**
c	Salida	PIN_H2	**HEX0[2]**
d	Salida	PIN_H1	**HEX0[3]**
e	Salida	PIN_F2	**HEX0[4]**
f	Salida	PIN_F1	**HEX0[5]**
g	Salida	PIN_E2	**HEX0[6]**

Tabla 1. Asignación de patillas de la FPGA a las señales del circuito.

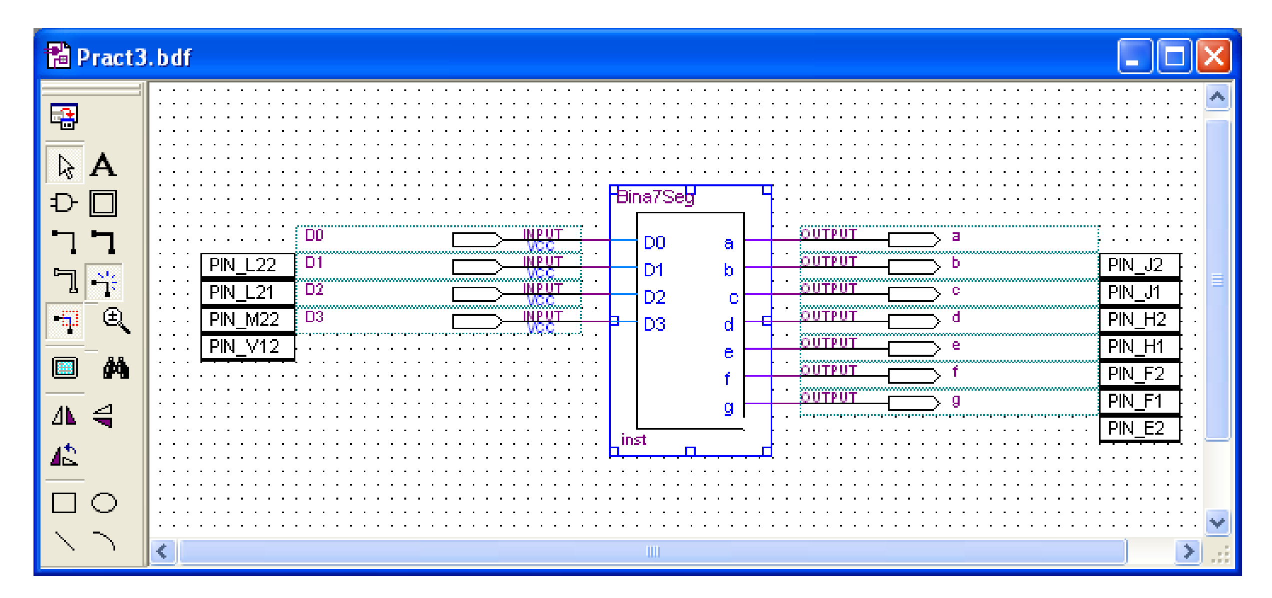


Figura 30. Nivel superior de la jerarquía con asignación de patillas.

CERRAR EL PROGRAMA

Una vez asignadas las patillas a todos los terminales, hay que salir del programa seleccionando la opción `Exit` del menú `File`, guardando los cambios en el caso de que se haya realizado alguna modificación de última hora.

Es imprescindible realizar una copia de seguridad del directorio de la práctica (`c:\Temp\Practica`), ya que el contenido del directorio temporal del ordenador puede ser borrado por otro usuario.

ANEXO A

